A Multiple-Model Based Framework for Automatic Speech Segmentation

Seung Seop Park, Jong Won Shin, Jong Kyu Kim and Nam Soo Kim

School of Electrical Engineering and INMC
Seoul National University, Korea
{sspark, jwshin, ckkim}@hi.snu.ac.kr, nkim@snu.ac.kr

Abstract

We propose a new approach to automatic speech segmentation for corpus-based speech synthesis. We utilizes multiple independent automatic segmentation machines (ASMs), instead of using a single ASM, to get final segmentation results: Given multiple independent time-marks from various ASMs, we remove biases of the time-marks, and then compute the weighted sum of the bias-removed time-marks. The bias and weight parameters needed for the proposed method are estimated for each phonetic context through a training procedure where manually-segmented results are used as the references. The bias parameters are obtained by averaging the corresponding errors. The weight parameters are simultaneously optimized through the gradient projection method to overcome a set of constraints in the weight parameter space. A decision tree is employed to deal with the useen phonetic contexts. Experimental results show that the proposed method remarkably improves the segmentation accuracy.

Index Terms: Automatic speech segmentation, speech synthesis, unit selection.

1. Introduction

The unit-selection based text-to-speech (TTS) synthesis system [1] largely depends on a speech corpus itself to achieve high-quality synthetic speech. One of the key tasks for building a corpus is to mark the boundaries of each speech segment according to the given phonetic transcript. Although manual segmentation is considered the most reliable and precise way to get the boundary locations, it requires a lot of time and labor. Therefore, an automatic segmentation method is considered more desirable and practical especially when the size of corpus is large.

In the literature, a variety of approaches to automatic speech segmentation have been developed [2]-[6]. Most of the approaches are based on the hidden Markov model (HMM) which is widely used in the area of automatic speech recognition. It is generally known that various model configurations such as the number of states for each phone and the number of mixture components per state yield different segmentation results [2]. It is also known that the HMMs perform better for some transitions than for others, and make similar errors (or bias) for all the phonetically similar transitions [3]-[4].

The performance of an automatic HMM-based segmentation technique, however, is usually found insufficient to be directly applied to TTS. In order to overcome this limitation, various post-processing techniques have been developed to refine the boundaries of initial segmentation [3]-[4]. These methods are basically trying to minimize the gap between the (automatically determined) initial and (manually specified) target boundaries.

Recently, there have been several attempts to utilize multiple boundary time-marks obtained from a variety of segmentation methods to get a single final time-mark [5], [6]. The multiple time-marks are simply averaged in [5] while, in [6], a single (presumably the best) boundary time-mark among the ones provided by the multiple segmentation methods is selected depending on the phonetic context of the boundary. In this paper, we propose a general framework which extends the previous multiple-model based segmentation methods.

2. The proposed multiple-model based segmentation method

Let an automatic segmentation machine (ASM) be a general system that performs a segmentation task automatically, i.e., an ASM produces a sequence of boundary time-marks given an utterance and the corresponding phonetic transcript. We also define the boundary type (btype) for a boundary time-mark (bmark) as a pair of two phonetic identities adjacent to this time-mark. It is easy to expect that a phone boundary of an ASM is dominantly affected by the two adjacent (left and right) phonemes [6] and, thus, we assume that an ASM has a different performance for each btype. An ASM applies a specific algorithm to align an utterance along its phonetic labels. In this paper, however, our focus lies not on a specific algorithm of an ASM but on a general method regarding how to determine the boundary time marks in case multiple ASMs are available.

Suppose that there are available a number of, say N_a, ASMs which use a variety of algorithms of their own. Let us call these ASMs as the base ASMs. Given a speech signal and the corresponding phonetic transcript $\mathcal{P} = \{p_j\}_{j=0}^{L}$, the i-th base ASM produces a set of bmarks $\mathcal{T}_i = \{t_{j,i}\}_{j=1}^{L}$ where p_j is the phonetic identity, $t_{j,i}$ is the j-th bmark given by the i-th base ASM and L denotes the number of boundaries determined according to the transcript $\mathcal{P}$. The btype of bmark $t_{j,i}$ is defined as $\theta_j \equiv (p_{j-1}, p_j)$, which is independent of the ASM index i since all the base ASMs share the same phonetic transcript $\mathcal{P}$. Given the N_a base ASMs, our goal is to make a single, improved set of bmarks $\mathcal{T}_{\mathrm{F}} = \{t_{j,\mathrm{F}}\}_{j=1}^{L}$ based on all the bmark sets $\mathcal{T}_1, ..., \mathcal{T}_{N_a}$.

To achieve this goal some approaches have been investigated in the pervious studies. In [5], $t_{j,\mathrm{F}}$ is obtained by averaging the N_a base results, i.e., $t_{j,\mathrm{F}}^{\mathrm{avg}} = \frac{1}{N_a}\sum_{i=1}^{N_a} t_{j,i}$. On the other hand, in [6] the final boundary decision is made by selecting the best base ASM depending on the btype as follows: $t_{j,\mathrm{F}}^{\mathrm{sel}} = S_{\theta_j}(t_{j,1}, ..., t_{j,N_a})$ where S_{θ_j} is a mapping that chooses one of its arguments according to θ_j.

In this section, we propose a new method called the automatic segmentation by weighting multiple models with bias correction (ASWMBC) which is an extension of our previous

work presented in [6]. In the proposed method, we first remove the bias of each bmark provided by the base ASMs, and then compute the weighted sum to produce the final bmark. The parameters necessary for the bias removal and weighting are separately specified for each btype. Therefore, the j-th final bmark $t_{j,\mathrm{F}}$ is obtained as follows:

$$t_{j,\mathrm{F}}^{\mathrm{wmbc}} = \sum_{i=1}^{N_a} w_{\theta_j,i} \underbrace{(t_{j,i} - b_{\theta_j,i})}_{\equiv t'_{j,i}} \tag{1}$$

where $b_{\theta_j,i}$ and $w_{\theta_j,i}$ are respectively the bias and weight parameters for the i-th base ASM when the btype is θ_j, and $t'_{j,i}$ is the j-th bmark of the i-th base ASM after bias removal. Since multiplying a time by a negative value does not make sense and the final bmark $t_{j,\mathrm{F}}$ is desired to be confined in the region $\min_i t'_{j,i} \le t_{j,\mathrm{F}} \le \max_i t'_{j,i}$, we assume that the weights for any btype θ are constrained by

$$\sum_{i=1}^{N_a} w_{\theta,i} = 1 \quad \text{and} \quad w_{\theta,i} \ge 0 \ \text{ for } i = 1, ..., N_a. \tag{2}$$

Clearly, both the averaging and selection methods are nothing but special cases of the ASWMBC approach. If described in the ASWMBC framework, the averaging method corresponds to the case all the weight parameters are equal, while the selection method to the case that, for each btype, only one weight parameter (corresponding to the best ASM) is 1 and the others are 0.

3. Training of Bias and Weight Parameters

A set of manually-segmented data is used to train the weight and bias parameters. Let us assume that we have a manual segmentation data $\mathcal{T}_{\mathrm{M}} = \{t_{j,\mathrm{M}}\}_{j=1}^{L_{\mathrm{M}}}$, where L_{M} is the total number of the manual bmarks and $t_{j,\mathrm{M}}$ is the j-th manual bmark. Our goal is to train the bias and weight parameters such that they can minimize some distance between the manual bmarks and the output bmarks obtained from (1).

For notational brevity, we adopt the following notation:

$$\begin{aligned} \mathbf{b} &= [b_1, \ldots, b_{N_a}]^T, \\ \mathbf{w} &= [w_1, \ldots, w_{N_a}]^T, \quad \text{and} \\ \mathbf{e}_j &= [e_{j,1}, \ldots, e_{j,N_a}]^T \end{aligned}$$

in which $e_{j,i} = t_{j,i} - t_{j,\mathrm{M}}$ and T denotes matrix transpose. Let us denote the bias and weight DBs by $\mathcal{B} = \{\mathbf{b}_\theta\}_{\theta \in \mathbf{\Theta}_{\mathrm{M}}}$ and $\mathcal{W} = \{\mathbf{w}_\theta\}_{\theta \in \mathbf{\Theta}_{\mathrm{M}}}$, respectively, in which $\mathbf{\Theta}_{\mathrm{M}} = \bigcup_{j=1}^{L_{\mathrm{M}}} \theta_j$ is a collection of btypes observed in the training data, and $\mathbf{b}_\theta$ and $\mathbf{w}_\theta$ are respectively the bias and weight vectors for θ.

The overall distance between the manual and final segmentations for some $\mathcal{W}$ and $\mathcal{B}$ is given by

$$\begin{aligned} J_{\mathrm{total}}(\mathcal{W}, \mathcal{B}) &= \sum_{j=1}^{L_{\mathrm{M}}} f_c(e_{j,\mathrm{F}}(\mathbf{w}_{\theta_j}, \mathbf{b}_{\theta_j})) && (3) \\ &= \sum_{\theta \in \Theta_{\mathrm{M}}} \underbrace{\sum_{j \in \mathcal{I}_\theta} f_c(e_{j,\mathrm{F}}(\mathbf{w}_\theta, \mathbf{b}_\theta))}_{\equiv J_\theta(\mathbf{w}_\theta, \mathbf{b}_\theta)} && (4) \end{aligned}$$

where

$$e_{j,\mathrm{F}}(\mathbf{w}_\theta, \mathbf{b}_\theta) = t_{j,\mathrm{F}} - t_{j,\mathrm{M}} = \mathbf{w}_\theta^T(\mathbf{e}_j - \mathbf{b}_\theta), \tag{5}$$

and $f_c(\cdot)$ is a cost function which quantifies the difference between two bmarks. In (4), $\mathcal{I}_\theta = \{j : \theta_j = \theta \text{ for } 1 \le j \le L_{\mathrm{M}}\}$ is a set of indices for which the btype is θ in the training data, and $J_\theta(\mathbf{w}_\theta, \mathbf{b}_\theta)$ is the subcost for θ.

The optimal weight and bias parameters are estimated according to the following criterion:

$$\{\mathcal{W}^*, \mathcal{B}^*\} = \underset{\mathcal{W},\mathcal{B}}{\operatorname{argmin}} \ J_{\mathrm{total}}(\mathcal{W}, \mathcal{B}). \tag{6}$$

From (4), we can see that J_{total} is minimized when J_θ is minimized for each btype $\theta \in \mathbf{\Theta}_{\mathrm{M}}$. Therefore, the optimal estimation procedure can be carried out separately for each btype θ as follows:

$$\{\mathbf{w}_\theta^*, \mathbf{b}_\theta^*\} = \underset{\mathbf{w},\mathbf{b}}{\operatorname{argmin}} \ J_\theta(\mathbf{w}, \mathbf{b}). \tag{7}$$

Since, however, it is difficult to find $\mathbf{w}_\theta^*$ and $\mathbf{b}_\theta^*$ simultaneously, we apply a suboptimal approach where $\mathbf{b}_\theta^*$ is estimated first and it is held fixed while searching for $\mathbf{w}_\theta^*$, which will be described in the next two subsections.

Although any distance measure can be used for the cost function f_c, we adopt the traditional Euclidean metric, i.e.,

$$f_c(e) = e^2. \tag{8}$$

With this form of cost function, the subcost for θ in (4) is given by

$$J_\theta(\mathbf{w}, \mathbf{b}) = \mathbf{w}^T \mathbf{E}_\theta(\mathbf{b}) \mathbf{w} \tag{9}$$

in which

$$\mathbf{E}_\theta(\mathbf{b}) = \sum_{j \in \mathcal{I}_\theta} (\mathbf{e}_j - \mathbf{b})(\mathbf{e}_j - \mathbf{b})^T \tag{10}$$

is an error covariance matrix for btype θ. Without loss of generality, it will be assumed that (8) is applied as the cost function in the remaining of this paper.

3.1. Training of bias parameters

To find the optimal bias vector for btype θ, we solve the following equation for arbitrary $\mathbf{w}$:

$$\frac{\partial J_\theta}{\partial \mathbf{b}} = -2\mathbf{w}\mathbf{w}^T \sum_{j \in \mathcal{I}_\theta} (\mathbf{e}_j - \mathbf{b}) = \mathbf{0}, \tag{11}$$

yielding the optimal bias vector $\mathbf{b}_\theta^*$ given by

$$\mathbf{b}_\theta^* = \frac{1}{|\mathcal{I}_\theta|} \sum_{j \in \mathcal{I}_\theta} \mathbf{e}_j, \tag{12}$$

where $|\mathcal{I}_\theta|$ is the cardinality of $\mathcal{I}_\theta$. As seen from (12), the bias parameters are obtained by averaging the corresponding errors.

3.2. Training of weight parameters for boundary type θ

Once $\mathbf{b}_\theta^*$ has been found, the weight parameter vector $\mathbf{w}_\theta^*$ which minimizes J_θ in (9) is searched over the confined region shown in (2) with the bias vector held fixed to $\mathbf{b}_\theta^*$. To accomplish this goal, we apply the gradient projection (GP) method, which is suitable for solving constrained optimization problems [7]. At each iteration, we find a feasible direction by projecting the negative gradient of the objective function onto the tangent subspace specified by an active set of constraints, and as moving toward this direction, we find the minimal point which becomes the next starting point. Detailed procedures for the GP method specific to our case are given as follows:

0) Initialize the weight vector $\mathbf{w}$ such that it is feasible, and calculate the error covariance matrix $\mathbf{E}_\theta(\mathbf{b}_\theta^*)$.

1) Derive the index set $I = \{1 \le i \le N_a : w_i = 0\}$ and set $q = N_a - |I|$.

2) Find the feasible direction vector $\mathbf{d} = [d_1, ..., d_{N_a}]^T = -\mathbf{P}\nabla J_\theta(\mathbf{w})^T$ where $\mathbf{P}$ is a $N_a \times N_a$ projection matrix whose (i,j)-th component is given by

$$P_{ij} = \begin{cases} 0 & \text{if } i \in I \text{ or } j \in I \text{ or } q = 1 \\ 1 - 1/q & \text{else if } i = j \\ -1/q & \text{otherwise,} \end{cases}$$

and $\nabla J_\theta(\mathbf{w}) = 2\mathbf{w}^T\mathbf{E}_\theta(\mathbf{b}_\theta^*)$ is the gradient vector at $\mathbf{w}$.

3) If $\mathbf{d} \neq \mathbf{0}$,

a) calculate $\alpha_{\max}$ such that

$$\begin{aligned} \alpha_{\max} &= \max\left\{\alpha : \mathbf{w}^{(k)} + \alpha\mathbf{d}^{(k)} \text{ is feasible}\right\} \\ &= \min_{1 \le i \le N_a : d_i \neq 0} \left\{ \max\left\{ \frac{1 - w_i}{d_i}, -\frac{w_i}{d_i} \right\} \right\}, \end{aligned}$$

b) find α_{opt} such that

$$\begin{aligned} \alpha_{\text{opt}} &= \operatorname*{argmin}_{0 \le \alpha \le \alpha_{\max}} J_\theta(\mathbf{w} + \alpha\mathbf{d}) \\ &= \min\left\{ \max\left\{ 0, -\frac{\mathbf{w}^T\mathbf{E}_\theta(\mathbf{b}_\theta^*)\mathbf{d}}{\mathbf{d}^T\mathbf{E}_\theta(\mathbf{b}_\theta^*)\mathbf{d}} \right\}, \alpha_{\max} \right\}, \end{aligned}$$

c) set $\mathbf{w} = \mathbf{w} + \alpha_{\text{opt}}\mathbf{d}$ and return to 1).

4) If $\mathbf{d} = \mathbf{0}$, find λ_i for $i \in I$ where $\lambda_i = \frac{\partial J_\theta}{\partial w_i}\Big|_{\mathbf{w}} - \frac{1}{q}\sum_{1 \le j \le N_a : j \notin I} \frac{\partial J_\theta}{\partial w_j}\Big|_{\mathbf{w}}$.

a) If $\lambda_i \ge 0$ for all $i \in I$, stop the iteration with $\mathbf{w}_\theta^* = \mathbf{w}$.

b) Otherwise, delete $i_t = \text{argmin}_{i \in I}\lambda_i$ from I, set $q = q + 1$, and return to 2).

3.3. Clustering of Boundary Types

It is important for a robust estimation of the bias and weight parameters that there should be a sufficient amount of manually-segmented data for each btype. Since, however, the size of the manually-segmented data is not usually large, the training of the bias and weight parameters does not guarantee a reliable estimate for all the btypes. Furthermore, some btypes may not be even observed in the manually-segmented data. In order to apply the proposed method, we should also have the bias and weight parameters for those unseen btypes.

To overcome this difficulty, we employ a decision tree [8] which clusters all the btypes into a finite number of groups. The decision tree is built as follows: First, all boundaries of the manually-segmented data are pooled together at the root node of the tree. Then, this pool is subsequently split up into two child nodes according to phonetically-motivated questions, such as the place of articulation, the voicing of phone, and the preceding and following phonetic context of the boundary. The stopping criterion is to ensure at least δ data points in each leaf node.

At each node, the best question which split the node into two child nodes is chosen such that the sum of the two child nodes' sub-costs could be minimized, in which the sub-cost of a node is obtained by accumulating the errors given by eq. (5) in the node when the optimal bias and weight parameters estimated at the node are used.

Given the decision tree, all boundary types (including unseen ones) can be mapped to one of leaf nodes of the tree and the btypes reaching to the same leaf node share the same weight and bias parameters.

4. Experimental Results

The speech database we used consisted of 5000 Korean utterances (286082 phones) which were spoken by a professional female narrator in a studio environment and were recorded in 16-bit precision with 16 kHz sampling frequency. In the speech database, manual segmentation results were available for 2000 utterances among which a maximum of 1600 utterances were used for training the bias and weight parameters and the remaining 400 utterances were reserved for performance evaluation.

In order to evaluate the performance of the proposed ASWMBC method, we implemented 36 base ASMs, all were built based on the HMM approach. To train the HMM-based ASMs, a feature vector was extracted for each frame with 24 ms window length and 3 ms frame shift. The feature vector was composed of 12 MFCCs, normalized log energy, and their first and second order delta components (39-dimension in total). The basic structure of the phone HMMs was a left-to-right type without any state skipping. In addition, the observation distribution specified in each state was characterized by the Gaussian mixture model with a finite number of mixture components. The 36 base ASMs were established by varying the manner of incorporating context dependency in the HMM (context-independent or -dependent model), the number of states for each phone HMM (3, 4, or 5 states) and the number of mixture components per each state ($1 \sim 6$ mixtures). Each combination of these HMM configurations gave us an unique base ASM. All the HMMs were trained over the 4600 utterances (excluding the evaluation data) without any manual segmentation information by means of HTK [9] software, where state tying was applied to estimate the parameters of the context-dependent triphone models.

Given the 36 base ASMs, 400 manually-segmented utterances were used to train the bias and weight parameters for the proposed ASWMBC method. There were 949 btypes observed in the training database, while 1218 btypes existed in the entire database. To cope with the unseen btypes and to train the parameters in a robust way, a decision tree was grown to cluster all the btypes in the training data as described in Subsection 3.3. The stopping criterion was to ensure that there should be at least δ data points for each leaf node of the tree, yielding 499 leaf nodes for $\delta = 10$. Each btype observed in the entire database could be mapped to some leaf node by the decision tree, and the bias and weight parameters were calculated by utilizing the manually-segmented data of the leaf node. In this way, the bias (or weight) DB which defines an one-to-one mapping from a btype to a bias (or weight) vector was constructed.

After the decision tree had been built, a variety of bias and weight DBs were created for performance comparison. We constructed two bias DBs, $\mathcal{B}^*$ and $\mathcal{B}'$, and five separate weight DBs, $\mathcal{W}_{\text{opt}}^*$, $\mathcal{W}_{\text{opt}}'$, $\mathcal{W}_{\text{sel}}^*$, $\mathcal{W}_{\text{sel}}'$ and $\mathcal{W}_{\text{avg}}$. $\mathcal{B}^*$ is the optimal bias DB obtained by the error averaging procedure as described in subsection 3.1, while $\mathcal{B}'$ is a null bias DB which always outputs a zero vector for any btype. The null bias DB was applied to discriminate the performance improvement due to the bias correction scheme from that due to the weighting scheme in the ASWMBC method. On the other hand, the various weight DBs were made in order to compare the optimal weight parameters obtained based on the proposed GP method with those based

Table 1: Performances of the ASWMBC approach for various weight and bias DBs. (trained with 400 utts. and $\delta = 10$)

bias DB	weight DB	MAE (ms)	RMSE (ms)	<20ms (%)
$\mathcal{B}'$	best single ASM	9.78	14.83	88.01
$\mathcal{B}'$	$\mathcal{W}_{\mathrm{avg}}$	9.59	13.10	89.19
$\mathcal{B}'$	$\mathcal{W}'_{\mathrm{sel}}$	8.44	13.31	90.65
$\mathcal{B}'$	$\mathcal{W}'_{\mathrm{opt}}$	7.78	11.86	92.26
$\mathcal{B}^*$	best single ASM	7.36	12.72	91.70
$\mathcal{B}^*$	$\mathcal{W}_{\mathrm{avg}}$	6.58	10.83	93.54
$\mathcal{B}^*$	$\mathcal{W}^*_{\mathrm{sel}}$	6.67	11.73	93.18
$\mathcal{B}^*$	$\mathcal{W}^*_{\mathrm{opt}}$	6.20	10.57	94.32

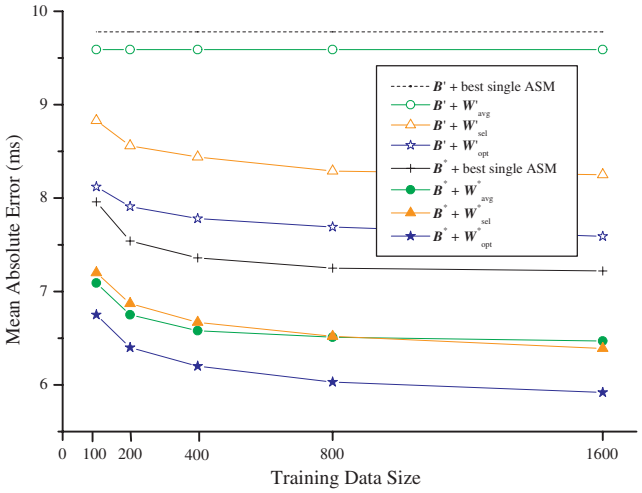

Figure 1: Mean absolute error for various bias and weight parameters as the number of training data is varying.

on the previous averaging and selection methods. $\mathcal{W}^*_{\mathrm{opt}}$ (or $\mathcal{W}'_{\mathrm{opt}}$) and $\mathcal{W}^*_{\mathrm{sel}}$ (or $\mathcal{W}'_{\mathrm{sel}}$) consists of the optimal and selection weights, respectively, trained after removing bias with $\mathcal{B}^*$ (or $\mathcal{B}'$), and $\mathcal{W}_{\mathrm{avg}}$ always gives the averaging weight for any btype.

Once both the bias and weight DBs had been specified, the final segmentation results were obtained by applying (1) with the DBs. The performances against various combinations of the bias and weight DBs are shown in Table 1, where we measured the performances in terms of the mean absolute errors (MAE), the root mean square errors (RMSE) and the percentages of boundaries deviating less than 20 ms from the manual boundaries. For the purpose of comparison, the performance of the best base ASM before/after bias correction is also presented. Among the results shown in Table 1, the use of both $\mathcal{B}^*$ and $\mathcal{W}^*_{\mathrm{opt}}$ achieved the best performance for all figures of merits, which means that the proposed method worked better than the previous averaging and selection methods in this experiment.

Fig. 1 shows how the performance (measured in terms of MAE) is affected by the amount of training utterances. In overall, the performance of each method became improved as more training data were used. We also conducted additional experiments in which δ varied from 10 to 50, and the experimental result demonstrated that the performance is somewhat insensitive to the choice of δ. As we can see in Fig. 1, the optimal weights trained by the GP algorithm outperformed the averaging and selection weights for *all* training conditions. This encouraging results lead us to convince the superiority of the proposed method over the averaging and selection methods.

5. Conclusions

In this paper, we have proposed an automatic speech segmentation method based on multiple ASMs. In the proposed method, given multiple boundary time-marks provided by various independent segmentation methods, a single final time-mark is obtained by weighted-averaging these time-marks after the bias of each time-mark is compensated. The bias and weight parameters are estimated for each boundary type through a training procedure, in which the optimal bias and weight parameters are estimated through the error averaging procedure and gradient projection method, respectively. We also employed a decision tree in order to deal with the boundary types unseen in the training data. Through various experiments, it was confirmed that the proposed method not only improves the segmentation accuracy considerably but also outperforms the previous multiple-ASM based approaches such as the averaging and selection methods.

6. References

[1] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, Atlanta, USA, pp. 373-376, 1996.

[2] H. Kawai and T. Toda, "An evaluation of automatic phone segmentation for concatenative speech synthesis," in *Proc. ICASSP*, vol. I, Montreal, Canada, pp. 677-680, 2004.

[3] J. Matoušek, D. Tihelka, and J. Psutka, "Automatic segmentation for czech concatenative speech synthesis using statistical approach with boundary-specific correction," in *Proc. Eurospeech*, Geneva, Switzerland, pp. 301-304, 2003.

[4] D.T. Toledano, L.A.H. Gómez, and L.V. Grande, "Automatic phonetic segmentation," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 6, pp. 617-625, Nov. 2003.

[5] J. Kominek and A. W. Black, "A family-of-models approach to HMM-based segmentation for unit selection speech synthsis," in *Proc. ICSLP*, Jeju, Korea, 2004.

[6] S. S. Park and N. S. Kim, "Automatic segmentation based on boundary-type candidate selection," *IEEE Signal Processing Letters*, vol. 13, no. 10, pp. 640-643, Oct. 2006.

[7] D. Luenberger, *Linear and Nonlinear Programming (Second Edition)*, pp. 330-334, Addison-Wesley, 1984.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. New York: Chapman & Hall, 1984.

[9] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2)*, Cambrige University Engineering Department, 2002.